



[40101/00401]

IN THE UNITED STATES PATENT AND TRADEMARK OFFICEAFS
IRW

Applicants : Roger WILES et al
Serial No. : 09/738,787
Filed : February 22, 2001
For : SYSTEM AND METHOD FOR UNIFORM
ACCESS OF MANAGED DATA OBJECTS
Group Art Unit : 2126
Examiner : Van H. Nguyen

Mail Stop: Appeal Brief - Patent
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Certificate of Mailing

I hereby certify that this correspondence is being
deposited with U.S. Postal Services as priority mail in an envelope
addressed to:

Mail Stop: Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

By:

Michael J. Marcin, (Reg. No. 48,198)

Date: July 19, 2004

TRANSMITTAL

S I R:

Transmitted herewith please find three (3) copies of an Appeal Brief for filing in the
above-identified application.

The Commissioner is hereby authorized to charge the Appeal Brief filing fee of \$330
and any additional fees required to the Deposit Account of **Fay Kaplun & Marcin, LLP No. 50-
1492**. A copy of this Transmittal is enclosed for that purpose.

Respectfully submitted,

Dated: July 19, 2004

By:

Michael J. Marcin, Reg. No. 48,198

Fay Kaplun & Marcin, LLP
150 Broadway, Suite 702
New York, NY 10038
Tel: (212) 619-6000
Fax: (212) 619-0276



PATENT
Attorney Docket No.: 40101-00401

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES**

In re Application of:)	
)	
Roger Wiles, et al.)	
)	
Serial No.: 09/738,787)	Group Art Unit: 2126
)	
Filed: December 15, 2000)	Examiner: Van H. Nguyen
)	
For: SYSTEM AND METHOD FOR)	Board of Patent Appeals and
UNIFORM ACCESS OF)	Interferences
MANAGED DATA OBJECTS)	

Commissioner for Patents
P.O. Box 1450
Arlington, VA 22313-1450

Sir:

APPEAL BRIEF UNDER 37 C.F.R. § 1.192

In support of the Notice of Appeal filed May 21, 2004, and pursuant to 37 C.F.R. § 1.192, Appellants present in triplicate their appeal brief in the above-captioned application.

This is an appeal to the Board of Patent Appeals and Interferences from the Examiner's final rejection of claims 1, 4-10, 12, 14, and 16-18 in the final Office Action dated January 29, 2004 as maintained in the Advisory Action mailed July 6, 2004. The appealed claims are set forth in the attached Appendix A.

1. Real Party in Interest

This application is subject to an assignment from the inventors to Wind River Systems Inc. of Alameda, California, the real party in interest.

2. Related Appeals and Interferences

There are no other appeals or interferences which would directly affect, be directly affected, or have a bearing on the instant appeal.

3. Status of the Claims

Claims 1, 4-10, 12, 14, and 16-18 are pending.

Claims 1, 4-10, 12, 14, and 16-18 were rejected in the Final Office Action and are involved in this appeal.

4. Status of Amendments

All amendments submitted by the appellants have been entered. None were submitted after the Advisory Action.

5. Summary of the Invention

The present invention is directed to a method comprising the steps of creating a producer component including a data object and a component module, the component module including information identifying the data object and an object handler to interact with the data object, registering the component module with an intermediary module, providing from a consumer component to the intermediary module a request for the data object in combination with the steps of correlating the requested data object with the component module which includes the

requested data object using the identifying information in the component module, forwarding the request to the component which interacts with the data object through the object handler, and fulfilling the request by providing the requested data object to the consumer component.

Figures 5a and 5b show an exemplary process for generating an IDB+ module 16 (e.g., component module) according to the present invention. Generation of the IDB+ module 16 includes a step 61 of obtaining a MIB file 31 (e.g. Management Information Base file), a step 63 of checking the MIB file 31 for syntax errors, and a step 65 of creating a IDB+ configuration file 32. *Specification*, page 13, lines 16-30. The process further includes a step 68 of generating a module source file 33 and module include file 34, the module source file 33 including a translation table 24, an object descriptor table 22, and an object ID table 23. *Specification*, page 14, lines 22-30. In a step 71, the module source file 33 and the module include file 34 are linked and compiled to generate the IDB+ module 16, and the IDB+ module 16 is tested. *Specification*, page 16, lines 28-32. The IDB+ module 16 may be registered with an IDB+ engine 11 according to the registration procedure for the IDB+ module shown in Figure 6.

Figure 7 shows an exemplary method according to the present invention for fulfilling a request (e.g. a Get request) from a consumer component 20 for a particular data object 17. In step 85, the request from the consumer component 20 is received by a dispatch interface component 14 of an IDB+ engine 11 of an IDB+ component 10. *Specification*, page 19, lines 2-4.

In a step 89, the dispatch interface component 14 initiates a search for the requested data object 17 and, in a step 91, checks for an alternate data object (e.g. a first data object 17 supported by another data object 17). *Specification*, page 20, lines 1-15. In a step 93 shown in Figure 7, the additional request addressed to the alternate data object may be received and processed. In a step 95, an IDB_t record is generated for the request, including information regarding the data object 17 (e.g., local ID, flags, index, etc.). *Specification*, page 21, lines 12-17. In a step 97, the dispatch interface component 14 calls an object handler routine in the corresponding IDB+ module 16 to implement the request in regard to the requested data object 17 based on information stored in the IDB_t record. *Specification*, page 21, lines 27-32.

The present invention is also directed to an intermediary module for a software package for facilitating communication among a plurality of components of a computing system, comprising a component module including information identifying a first one of the components and an object handler to interact with a data object, the first one of the components including the data object, a register configured to register the component module, and a dispatch component to route a request for the data object received from a second one of the components, the dispatch component correlating the requested data object to the component module including the requested data object, the correlation including the generation of a record including at least a portion of the identifying information included in the component module.

Figure 2 shows an example of an IDB+ component 10 according to the present invention. The IDB+ component 10 may manage requests submitted by a consumer component 20 to be fulfilled by a producer component 30. *Specification*, page 5, lines 1-3. Shown in Figure 3, the IDB+ component 10 may include control data 26 comprising a module hash table 28 and an object has table 29. *Specification*, page 10, lines 4-8. The module hash table 28 includes a list of every IDB+ module registered with the IDB+ component 10. *Id.* The IDB+ component 10 may also include a dispatch interface component 14 which routes the consumer component 20 request to a corresponding one of the IDB+ modules 16. *Specification*, page 9, lines 25-27.

Figure 2 additionally shows the IDB+ module 16 and the data object 17 according to the present invention. The IDB+ module 16, shown in more detail in Figure 4, is an individual manageable entity supporting data access to a representative data object 17. *Specification*, page 10, lines 30-31. The IDB+ module 16 may include a local ID enumeration table 21, a translation table 24, an object descriptor table 22, and an object ID table 23. *Specification*, page 10, lines 31-32; page 11, lines 1-4. The local ID enumeration table 21 includes a list of local IDs for the data object 17. *Id.* The object ID table 23 has a predefined data structure including information about the data object 17 including, for example, a local ID, a size of the data object 17, a valid range of values for the data object 17, a type of the data object 17, access rights, a flag indicating the status of the data object 17, a pointer to the name string of an alternate data object, etc. *Specification*, page 11, lines 15-25.

6. Issues

I. Whether claims 1, 4-10, 12, 14, and 16-18 are unpatentable under 35 U.S.C. § 103(a) as obvious over U.S. Patent No. 6,085,030 to Whitehead et al. ("*Whitehead*").

7. Grouping of Claims

Claims 1, 4-10, 12, 14, and 16-18 stand together.

8. Argument

I. The Rejection of Claims 1, 4-10, 12, 14, and 16-18 Under 35 U.S.C. § 103(a) as Unpatentable Over U.S. Patent No. 6,085,030 to Whitehead et al. Should be Reversed.

The Examiner has rejected each of the pending claims under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent 6,085,030 to Whitehead et al. ("*Whitehead*"). *Office Action*, paragraph 4. *Whitehead* discloses a component server architecture that enables consumer nodes of a network to interact with software components and services throughout the network. The interaction between the consumer nodes and various application is implemented by registering and locating requested components and services via a component registry. *Whitehead*, Abstract.

Independent claim 1 of the present invention recites a method comprising a first step of "creating a producer component including a data object and a component module, the component module including information identifying the data object and an object handler to

interact with the data object.” As described in the specification, each producer component includes a data object and a component module (*e.g.*, an IDB+ module). The IDB+ module is generated to include information about the data objects and to include an object handler to interact with the data object. Claim 1 further recites “registering the component module with an intermediary module.” Thus, it is the IDB+ module which is registered with the intermediary module.

In contrast, *Whitehead* does not teach nor disclose, either expressly or impliedly, the creation of a component module, which includes information identifying a data object, and an object handler to interact with a data object. Nor does *Whitehead* teach or disclose the registering of such a component module with an intermediary module. Rather, *Whitehead* teaches that the components (*i.e.* data objects) are directly registered with the component registry. *Whitehead*, col. 8, lines 6-9.

The creation of the component module and the registration of the component module with the intermediary module in the present invention adds a layer of abstraction which allows the consumer components transparent access to the data objects. In addition, the “information identifying the data object” in each component module operates to simplify a request for the data object. The information identifying the data object, as described in the specification, may include a flag indicating the status of a data object (*e.g.* valid or invalid), a pointer to the

name string of the data object, and a pointer to the name string of an alternate data object. This information, unique to each data object, provides an advantage in the step in claim 1 of “correlating the requested data object with the component module which includes the requested data object.” For example, a request for a first data object that is supported by another data object (e.g., alternate data object) may result in the request locating both the first data object and the predetermined alternate data object, thus simplifying the request and avoiding unnecessary delays. *Specification*, page 11, 17-32; page 12, 1-3.

In the Final Office Action, the Examiner suggested that the description of component server containers having “additional attributes and properties to specify configuration information and persistent data for the named component server” is equivalent to the component module according to present invention. *Final Office Action*, page 3, section 5. The Appellants respectfully disagree with the Examiner’s suggestion, and submit that the above reference fails to illustrate usage of a component module to provide consumer components with requested data objects. Lacking disclosure in the prior art, the component module (e.g., IDB+ module) of the present invention is an individual manageable entity supporting data access to a representative data object. *Specification*, page 10, line 30-31. *Whitehead* fails to disclose such component modules or a component architecture wherein component modules are used as intermediaries between components (i.e., consumers and producers).

The Appellants further submit that *Whitehead* specifically teaches away from the use of an intermediary, such as the component module, and discloses only direct interaction between consumer and producer components. A *prima facie* case of obviousness may be rebutted by showing that the art, in any material respect, teaches away from the claimed invention. *In re Geisler*, 116 F.3d 1465,1471,43 USPQ2d 1362, 1366 (Fed.Cir.1997). In *Whitehead*, consumer requests are sent to a component management service which are then forwarded to a global component registry to locate the requested object, where “the component registry is the single point of information for routing of all software component requests from consumers.” *Whitehead*, col. 10, lines 16-18 (emphasis added). The component registry of *Whitehead* stores information about producer objects and not any intermediate components. *Whitehead*, col. 10, lines 16-40. After the producer object is located, a reference to the producer is returned to the consumer directly. *Whitehead*, col. 8, lines 32-35. In contrast, claim 1 of the present invention includes a step of “forwarding a request to the component module which interacts with the data object component modules including information identifying the data object.” Each component module, as shown in Figure 4, is associated with a single data object including identifying information for that data object. The request is then implemented by an object handler in the corresponding component module.

The present invention additionally relies on component modules to provide uniform access to data objects which is not possible with direct consumer-to-producer

architecture disclosed in *Whitehead*. If the data object is modified or updated, those changes must be tracked and synchronized by multiple consumer components in the system. In certain instances, the particular network consumer component may not have an application programming interface (API) suited to access a particular desired producer data object. This problem is described in *Whitehead*:

If the object models of the offer or implementation match the [native component object] NCO model 220 of the requesting application, the object reference is returned via paths 22 and 23; otherwise, the interface adapter repository 270 is used to bridge the object models via paths 16 and 19, as described further with reference to FIG. 6. Briefly, the interface adapter 270 sends a request to the interface adapter repository 256 (over paths 17 and 18 of CMS 280) to build an appropriate interface between the offered component and the object model of the requesting application. As a result, an offer is returned to the requesting application (via paths 22 and 23) that resembles a native component.

Whitehead, col. 8, lines 32-43.

The present invention, with the use of component modules simplifies the management of data objects in such situations. Even if the data object does not match the NCO model of the requesting application, the component module encapsulation will act as an intermediary between the consumer and the producer objects, without going through the convoluted process disclosed in *Whitehead*.

Each novel component module of the present invention further includes “an object handler to interact with the data object” not taught or suggested in the prior art. “To support the conclusion that the claimed invention is directed to obvious subject matter, either the references

must expressly or impliedly suggest the claimed invention or the examiner must present a convincing line of reasoning as to why the artisan would have found the claimed invention to have been obvious in light of the teachings of the references.” *Ex parte Clapp*, 227 USPQ 972, 973 (Bd. Pat. App. & Inter. 1985). The teaching or suggestion to make the claimed combination must be found in the prior art and not based on applicant’s disclosure. *In re Vaeck*, 947 F.2d 488, 20 USPQ2d 1438 (Fed.Cir.1991). The Examiner admitted that *Whitehead* does not teach “an object handler to interact with a data object.” *Final Office Action*, page 4. However, the Examiner asserted that it would have been obvious to apply the teaching of the global component registry of *Whitehead* for an object handler “because it would have provided the capability for ensuring proper administration, authentication, and run-time binding access to components offered in response to requests from applications executing on the consumer nodes.” *Id.*

The Appellants respectfully disagree with the Examiner and submit that the Examiner provided a insufficient line of reasoning as to why the artisan would have found the claimed invention obvious in light of the teachings of the references. Further, the Appellants submit that the Examiner improperly relied on the Appellants disclosure to find the teaching or suggestion to make the claimed combination. As discussed above, the global component registry disclosed in *Whitehead* is “the single point of information for routing of all software component requests from consumers.” *Whitehead*, col. 10, lines 16-18 (emphasis added). In contrast, the object handler of the present invention is defined as a routine in a corresponding IDB+ module

(e.g., component module) used to implement a request based on information stored in an IDB_t record. *Specification*, page 23, lines 10-15; Figure 7. Independent claim 1 includes, for each data object, the creation of a “component module including information identifying the data object and an object handle to interact with the data object.” The use of such object handlers local to corresponding data objects is not obvious in light of the prior art’s teaching of a global component registry.

Whitehead neither teaches nor suggests “creating a producer component including a data object and a component module, the component module including information identifying the data object and an object handler to interact with the data object” as recited in claim 1. Similarly, claim 10 recites “a component module including information identifying a first one of the components and an object handler to interact with a data object, the first one of the components including the data object.” Claim 14 recites “each of the producer components including a data object and a component module, the component module including information identifying the data object and an object handler to interact with the data object.”

Therefore, at least for these reasons, it is respectfully submitted that all of the presently pending claims are allowable. Appellants respectfully submit that the Board overturn the Examiner’s rejection under 35 U.S.C. 103(a) of independent claims 1, 10 and 14 and all the claims depending therefrom.

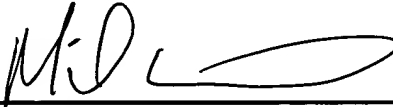
9. Conclusions

For the reasons set forth above, the appellants respectfully requests that the Board reverse the final rejections of claims 1, 4-10, 12, 14, and 16-18 by the Examiner under 35 U.S.C. §§ 103, and indicate that these claims are allowable.

Respectfully submitted,

FAY, KAPLUN & MARCIN, L.L.P.

Date: July 19, 2004

By: 
Michael J. Marcin
Reg. No. 48,198

Fay Kaplun & Marcin, LLP
150 Broadway, Suite 702
New York, New York 10038
Tel: (212) 619-6000
Fax: (212) 619-0276

APPENDIX A – APPEALED CLAIMS

1. A method, comprising the steps of:

creating a producer component including a data object and a component module,
the component module including information identifying the data object and an object handler to
interact with the data object;

registering the component module with an intermediary module;

providing from a consumer component to the intermediary module a request for
the data object;

correlating the requested data object with the component module which includes
the requested data object using the identifying information in the component module;

forwarding the request to the component module which interacts with the data
object through the object handler; and

fulfilling the request by providing the requested data object to the consumer
component.

4. The method according to claim 1, wherein the producer component is a hybrid component
which, under predetermined conditions, acts as a consumer component and which otherwise acts
as a producer component.

5. The method according to claim 1, wherein all of the components reside on a single processor.
6. The method according to claim 4, wherein the intermediary module receives a plurality of requests from the consumer component including at least one of a request to retrieve a value in the data object from the producer component, a request to retrieve a value in a next data object of the producer component, a request to set a value in the data object of the producer component, a request to set a read-only value of the data object of the producer component and a request to store a value of the data object in a nonvolatile memory.
7. The method according to claim 1, wherein the intermediary module performs the correlating step using one of a hash table, a database application and a binary tree.
8. The method according to claim 5, wherein the single processor operates a switching device.
9. The method according to claim 1, further comprising the step of de-registering the component module from the intermediary module.
10. An intermediary module for a software package for facilitating communication among a plurality of components of a computing system, comprising:

a component module including information identifying a first one of the components and an object handler to interact with a data object, the first one of the components including the data object;

a register configured to register the component module; and

a dispatch component to route a request for the data object received from a second one of the components, the dispatch component correlating the requested data object to the component module including the requested data object, the correlation including the generation of a record including at least a portion of the identifying information included in the component module.

12. The intermediary module according to claim 10, further comprising:

a configuration component including configuration parameters for the component module; and

a utility for generating the component module using the configuration component.

14. A system for managing communications among a plurality of components of a computing system comprising:

a consumer component;

a plurality of producer components, each of the producer components including a data object and a component module, the component module including information identifying the data object and an object handler to interact with the data object; and

an intermediary module receiving from the consumer component requests for data objects, wherein, upon receipt of a consumer component request, the intermediary module consults a register to identify the component module which includes the data to identify the requested data object.

16. The system according to claim 14, wherein the system operates a switch.

17. The system according to claim 14, wherein the intermediary module receives a plurality of requests from the consumer component including at least one of a request to retrieve a value in the a data object from the producer component, a request to retrieve a value in a next data object of the producer component, a request to set a value in the data object of the producer component, a request to set a read-only value of the data object of the producer component and a request to store a value of the data object in a nonvolatile memory.

18. The system according to claim 14, further comprising a hybrid component which, under predetermined conditions, acts as a consumer component and which otherwise acts as a producer component.